

NO-A191 247

DESIGN OF THE DETECTOR II: A CMOS GATE ARRAY FOR THE
STUDY OF CONCURRENT (U) STANFORD UNIV CA CENTER FOR
RELIABLE COMPUTING H A GOOSEN ET AL JUL 87

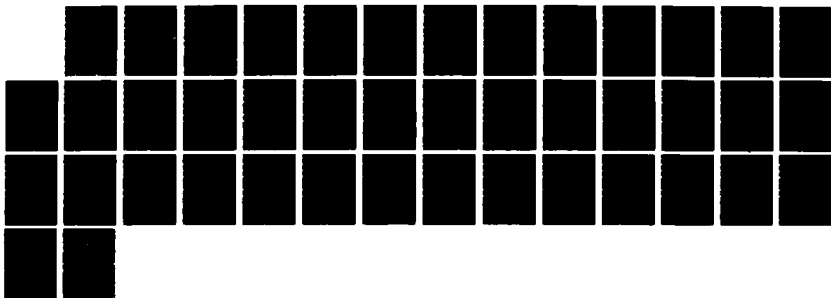
1/1

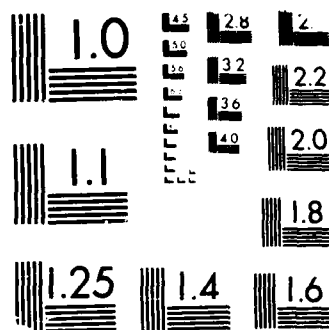
UNCLASSIFIED

CRC-TR-87-17 N00014-85-K-0600

F/G 9/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD-A191 247



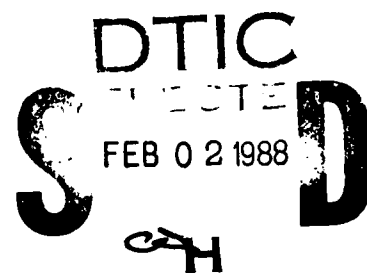
DTIC FILE COPY

**DESIGN OF THE DETECTOR II:
A CMOS GATE ARRAY FOR THE STUDY OF
CONCURRENT ERROR DETECTION TECHNIQUES**

Hendrik A. Goosen, Mario L. Côrtes and Edward J. McCluskey

CRC Technical Report No 87-17
CSL TN NO. 87-335

July 1987



Center for Reliable Computing
Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford CA 94305-4055 USA

Imprimatur: Hassanein H. Amer and Michael Parkin

This work was supported in part by the Council for Scientific and Industrial Research, South Africa, and in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization administered through the Office of Naval Research under contract No. N00014-85-K-0600.

Copyright © 1987 by the Center for Reliable Computing, Stanford University. All rights reserved, including the right to reproduce this report, or portions thereof, in any form.

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

**DESIGN OF THE DETECTOR II:
A CMOS GATE ARRAY FOR THE STUDY OF
CONCURRENT ERROR DETECTION TECHNIQUES**

Hendrik A. Goosen, Mario L. Côrtes and Edward J. McCluskey

CRC Technical Report No 87-17
CSL TN NO. 87-335

July 1987

Center for Reliable Computing
Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford CA 94305-4055 USA

Abstract

This report describes the *Detector II*, an experimental CMOS gate array circuit which was designed to study concurrent error detection schemes and temporary failures. The circuit consists of six different adders with concurrent error detection schemes. The error detection schemes are simple duplication, duplication with functional dual implementation, duplication with different implementations, two-rail encoding, low-cost residue coding, and parity prediction. Each adder contains circuitry which will be used to inject realistic temporary failures. Additional circuitry is provided to make selected internal nodes observable.



For	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	

TABLE OF CONTENTS

1. INTRODUCTION	1
2. PREVIOUS WORK	1
3. INJECTION OF TEMPORARY FAULTS	3
4. SIGNAL OBSERVATION	4
5. DESCRIPTION OF THE EXPERIMENT	4
6. DESIGN OF THE SYSTEM	5
7. DESIGN OF CED SCHEMES	7
7.1 Simple Duplication	7
7.2 Dual logic implementation	11
7.3 Alternative dual implementation	13
7.4 TSC two-rail adder	16
7.5 Parity prediction	16
7.6 Low cost residue coding	17
8. DESIGN OF SUPPORT CIRCUITRY	20
8.1 CED schemes tri-state buffers and latches	20
8.2 Counter	21
8.3 Buffer (CNTBUF)	21
8.4 Buffer (FINBUF)	21
8.5 Buffer (CTLBUF)	24
8.6 Buffer (CNTIB)	25
8.7 Reference adder	25
8.8 Comparator (SYSCMP)	27
ACKNOWLEDGEMENTS	27
REFERENCES	29
APPENDIX A: DRAWING BLOCK AND SIGNAL NAMES	30
APPENDIX B: OBSERVABLE INTERNAL NODES	32
APPENDIX C: FAULT INJECTION CONTROL	34
APPENDIX D: PACKAGE DETAIL	35

LIST OF FIGURES

1. Weak input fault injection	4
2. System structure	5
3. Support system	6
4. Adders with CED	6
5. Full schematic of the system	8
6. Simple duplication	9
8. Input buffer	10
7. 4-bit adder	11
9. 4-bit adder with fault injection	12
10. Dual full adder	13
11. TSC two-rail checker	13
12. 4-bit dual adder	14
13. Alternative dual full adder	15
14. Two-rail full adder	16
15. Duplicate carry unit	17
16. Adder with parity prediction	18
17. 4-bit adder with (mod 3) residue checking	19
18. (mod 3) adder	21
19. Output latches and tri-state buffers	22
20. 8-bit counter	23
21. Counter buffer	24
22. Final output buffer	24
23. Local fault injection signals input buffers	25
24. Test vector input buffer and global fault injection	25
25. Reference adder	26
26. System comparator	28

1. INTRODUCTION

For some applications of computer systems, errors have to be detected concurrently with normal operation. This is typically done by concurrent error detection (CED) circuits. Since about 90% of errors in computer systems are caused by temporary failures [McConnel 79], CED schemes have to effectively detect errors caused by temporary failures.

Most CED schemes [Wakerly 78], [Kraft 81] are designed with the assumption that errors are caused by events that can be modelled as single-stuck faults. There is a growing body of evidence which suggests that the single stuck-fault model does not model temporary failures very well [Cortes 87], [Amer 87].

This report describes the *Detector II*, a circuit which was designed to study concurrent error detection schemes experimentally. The purpose of the study is to find out how well the different schemes perform in the presence of real temporary failures, and to gain more knowledge of temporary failures in the process. This will also lead to better models for temporary failures.

The circuit was implemented as a CMOS gate array fabricated by Fairchild Gate Array, Milpitas, California. The circuit consists of approximately 2400 equivalent gates and is packaged in a 121 pin ceramic pin-grid array package.

2. PREVIOUS WORK

The central problems in the experimental investigation of error detection techniques are to inject the failures, and to observe the errors. The "failure generation" process must produce the same kind of errors one would expect from real physical failures. Similarly, the error observation procedure must allow one to determine unambiguously which errors were introduced, and how the system responded.

In fault simulation, faults are inserted into the system according to a fault model (such as the single-stuck model). The simulator then stores the response of the system. The same approach can be followed in experimental work. The validity of the results will then depend on the accuracy of the fault model.

[Crouzet 82] inserted permanent stuck-at faults into a microcomputer to evaluate its error detection mechanisms. Faults were injected into the microcomputer by a specially designed fault injector circuit. This circuit could place a stuck-at-1 and stuck-at-0 fault on every pin of a chip in the system. The system was then monitored to see whether or not it detected the injected fault, and what the effects of the fault were. An interesting note is that an unexpected fault turned up—a badly erased EPROM cell in one of the chips they tested. This fault was not modelled by a stuck-at fault, and was not detected by the detection mechanisms.

[Schuette 86] inserted temporary stuck-at faults into a microprocessor system to evaluate software CED schemes. A fault injection circuit inserted stuck-at faults on the processor bus. Insertion was done through an XOR gate located on each processor bus line. Fault duration could be set to one of three values: 1, 2, or 4 cycles.

In the previous two experiments, stuck faults were injected into the systems at the I/O pins. Recent experiments show that temporary failures often do not behave like stuck faults. [Cortes 87], [Cortes 86a], [Cortes 86b], [Cortes 86c] used power supply stress, extra loading on circuit nodes, and “weak input signals” to inject temporary and intermittent failures into TTL and CMOS circuits. [Amer 87] used low power supply voltage to inject temporary failures into a simple fault tolerant system. Both authors found evidence of faults that could not be explained by the stuck-fault model.

3. INJECTION OF TEMPORARY FAULTS

The experiments planned for the chip described in this report will improve on previous experimental studies of CED techniques by using the more realistic methods of fault injection described by C  rtes. Since the experiment will be performed on a specially designed CMOS VLSI chip, more specific information on temporary failures in CMOS will also be obtained.

The two most important fault injection techniques for this experiment will be power supply stress and weak input signals (described below). Other methods, such as electromagnetic interference, temperature stress, and electrostatic discharge are possible candidates for future experiments.

Power supply stressing of integrated circuits is described in [C  rtes 86a] and [C  rtes 86b]. In this technique, the power supply voltage to the system is reduced. A low power supply voltage reduces both the driving ability and the noise margins of logic gates. This causes delay faults and noise margin violations. C  rtes found that power supply stress caused intermittent faults in counter circuits.

The use of weak inputs is described in [C  rtes 87], and illustrated in Fig. 1. When a high signal is applied to the control pin, the target signal value passes through the AND gate to the next module. When a low signal is applied to the control pin, a stuck-at-0 fault is injected into the system. A weak input signal (voltage between the noise margins) on the control input causes the signal after the buffer to have an indeterminate value. This indeterminate value can propagate through the AND gate and result in an indeterminate value at its output. The target signal value may therefore be corrupted. The propagation of an indeterminate value is not well understood at the moment.

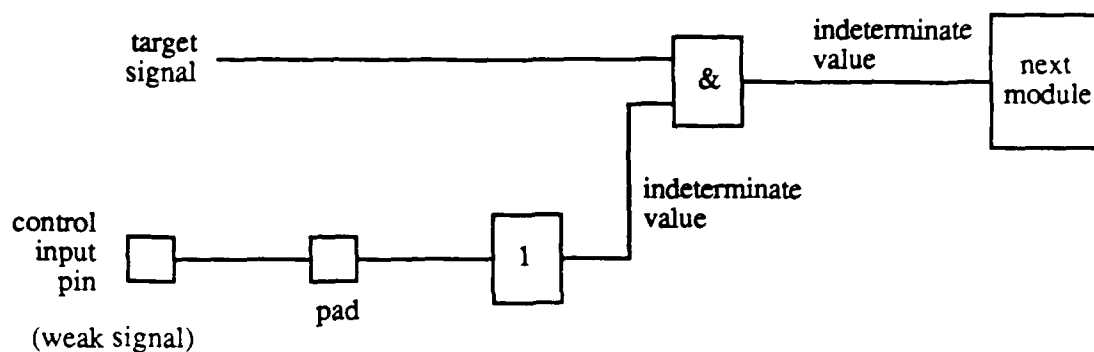


Figure 1. Weak input fault injection

4. SIGNAL OBSERVATION

The outputs of the CED circuits, as well as selected internal nodes, are buffered and connected to latches. Each latch samples the value of the node it is connected to, and in effect decides whether the node value is a one or a zero. This value is stable during the inactive clock phase.

5. DESCRIPTION OF THE EXPERIMENT

The circuits chosen for this experiment are simple 4-bit adders. Adders are used in many digital circuits. They are easy to test, and there are many documented techniques for detecting errors in adders. Six error detecting schemes were selected:

- simple duplication with matching by XOR gates
- duplicate and match using dual logic implementation—matching by two rail code TSC checkers
- duplicate and match using a “different dual” implementation
- two-rail adder with TSC checkers
- parity prediction
- residue coding.

6. DESIGN OF THE SYSTEM

The system was designed to be an evaluation tool. For that reason it includes circuitry to generate test patterns, inject faults, make internal nodes more observable, and monitor the experiment.

The structure of the system is shown in Fig. 2. It consists of two separate subsystems with no on-chip interconnection. This arrangement allows for the separation of the stress applied to the circuit under test from the test vector generation and the observation of the experiment. The intention is to use one copy of the chip for controlling the experiment, while faults are injected into another copy.

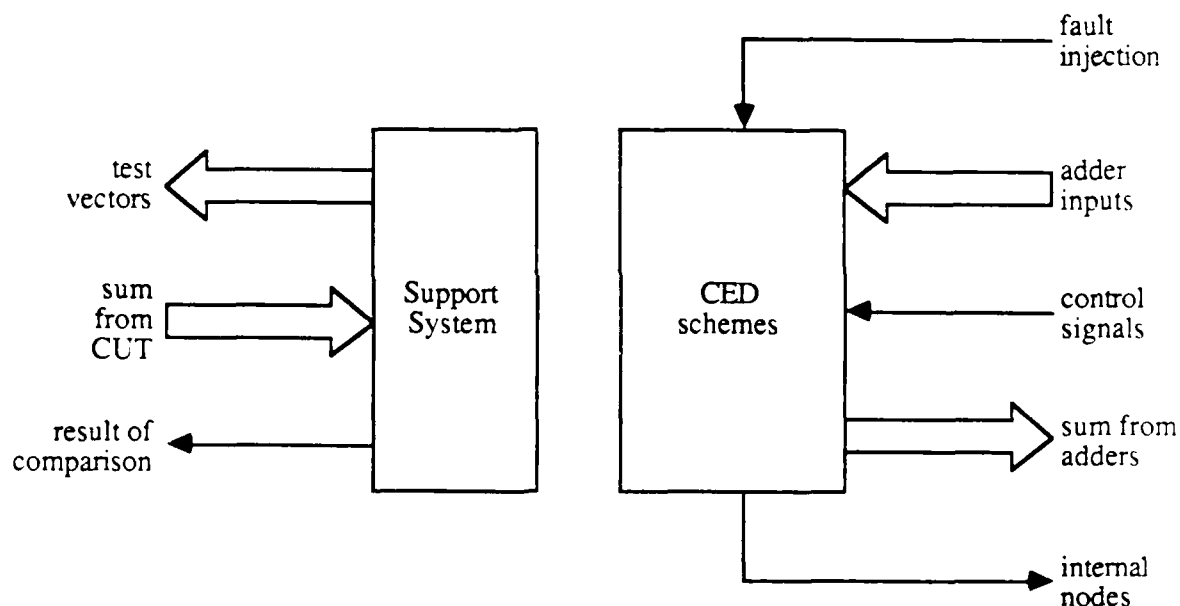


Figure 2. System structure

The support system is shown in Fig. 3. It consists of an 8-bit counter, a 4-bit reference adder, and a comparator. The counter generates exhaustive test patterns for the stressed adders. The counter output is connected to the reference adder, and also to output buffers. The reference adder generates the fault-free response to the test patterns. The comparator compares this to the output of the circuit under test (CUT).

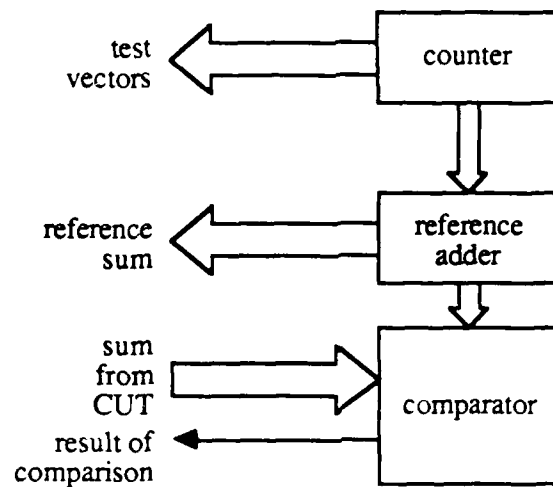


Figure 3. Support system

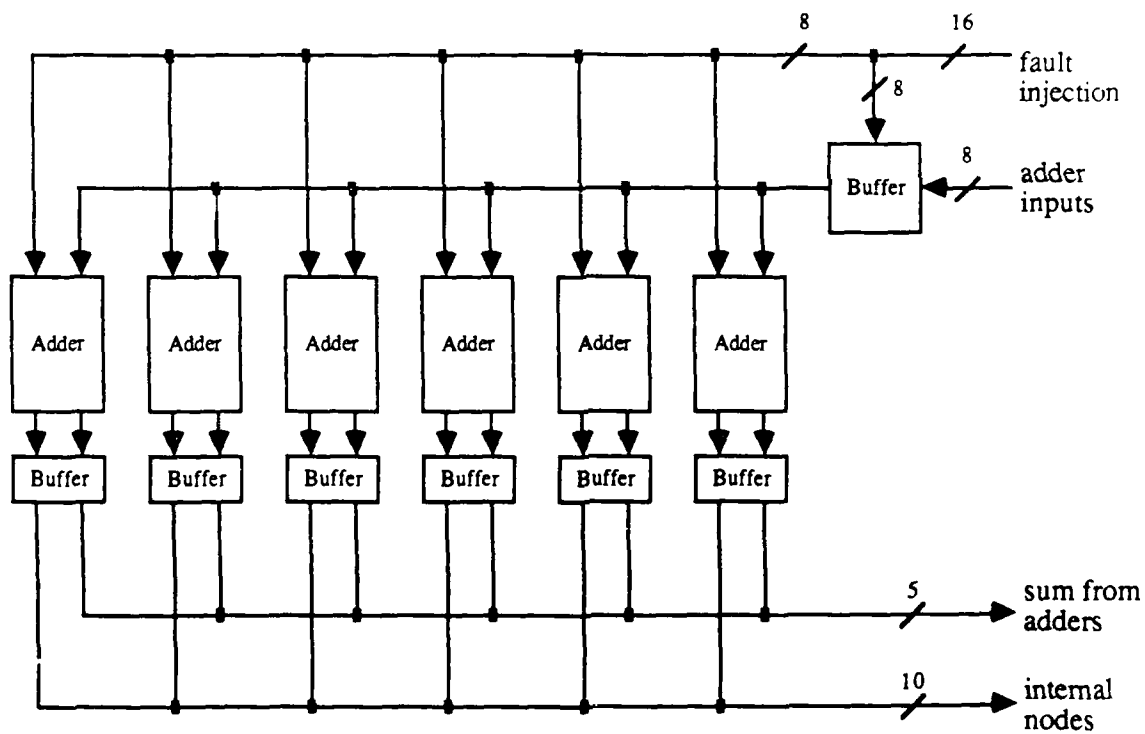


Figure 4. Adders with CED

The adders with CED are shown in Fig. 4. The data inputs of the six 4-bit adders are connected to two 4-bit wide data buses. Faults can be injected into the bus lines through

circuitry in the data bus input buffer (this is referred to as global fault injection). Fault injection directly into the adders (local fault injection) is controlled by the local control bus. All the adder outputs are latched and connected to an output bus through tristate buffers. Several internal nodes in each adder are made observable as shown. There is a tradeoff here between the amount of extra information made available, and the cost in extra output pins. It was decided (rather arbitrarily) to observe ten nodes in each adder. Each of the chosen nodes are connected to a buffer which drives a latch. This ensures that the value of the node is sampled every clock cycle while there is little extra loading on the node. Since some of the adders have two-rail outputs, all the adders provide both true and complemented error signals. This allows for more uniformity in the design.

The full top-level schematic of the system is shown in Fig. 5. An explanation of all the block and signal names can be found in Appendix A. We will now discuss each of the schemes in detail.

7. DESIGN OF CED SCHEMES

7.1 Simple Duplication

This is a system level technique in which the logic is duplicated, and XOR gates are used to compare the outputs of the two circuits [Carter 64]. One of the circuits is used to provide the system output, while the other is used for checking purposes only. Disagreement between the two circuits is detected by an array of XOR gates, and an error is signalled.

The circuit is shown in Fig. 6. The two function blocks CTLADD and ADD4 are the 4-bit adders. ADD4 is a simple 4-bit adder with ripple carry (shown in Fig. 7). Each of the blocks labelled ADD01* in Fig. 7 represents a full adder. CTLADD is a 4-bit adder that has been modified for fault injection and observation of internal nodes. The internal detail is shown in Fig. 9 and discussed below. The inputs to CTLADD and ADD4 are buffered to reduce the

* ADD01 is the name of a Fairchild gate array "macro" which implements the functionality of a full adder.

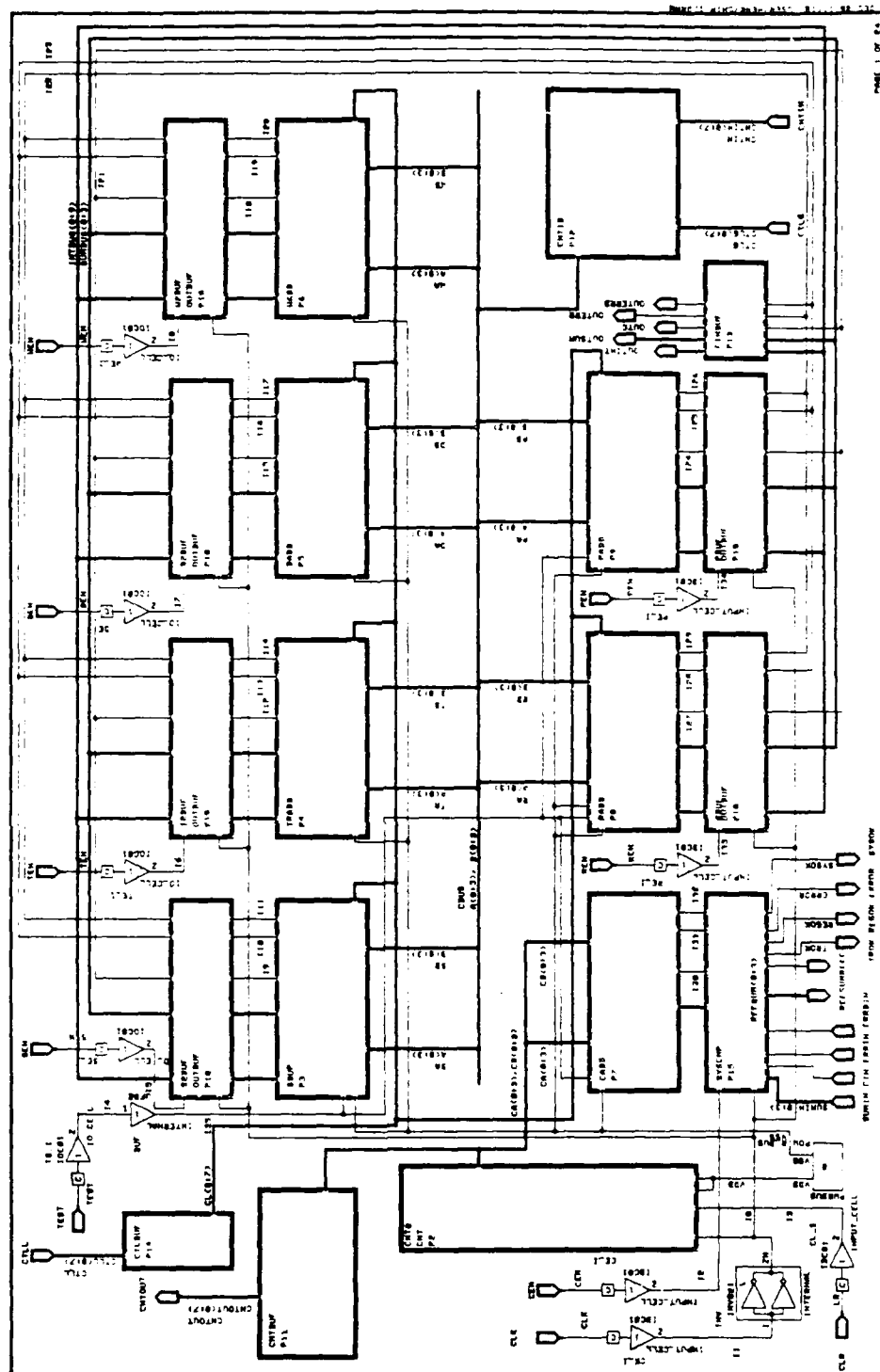


Figure 5. Full schematic of the system

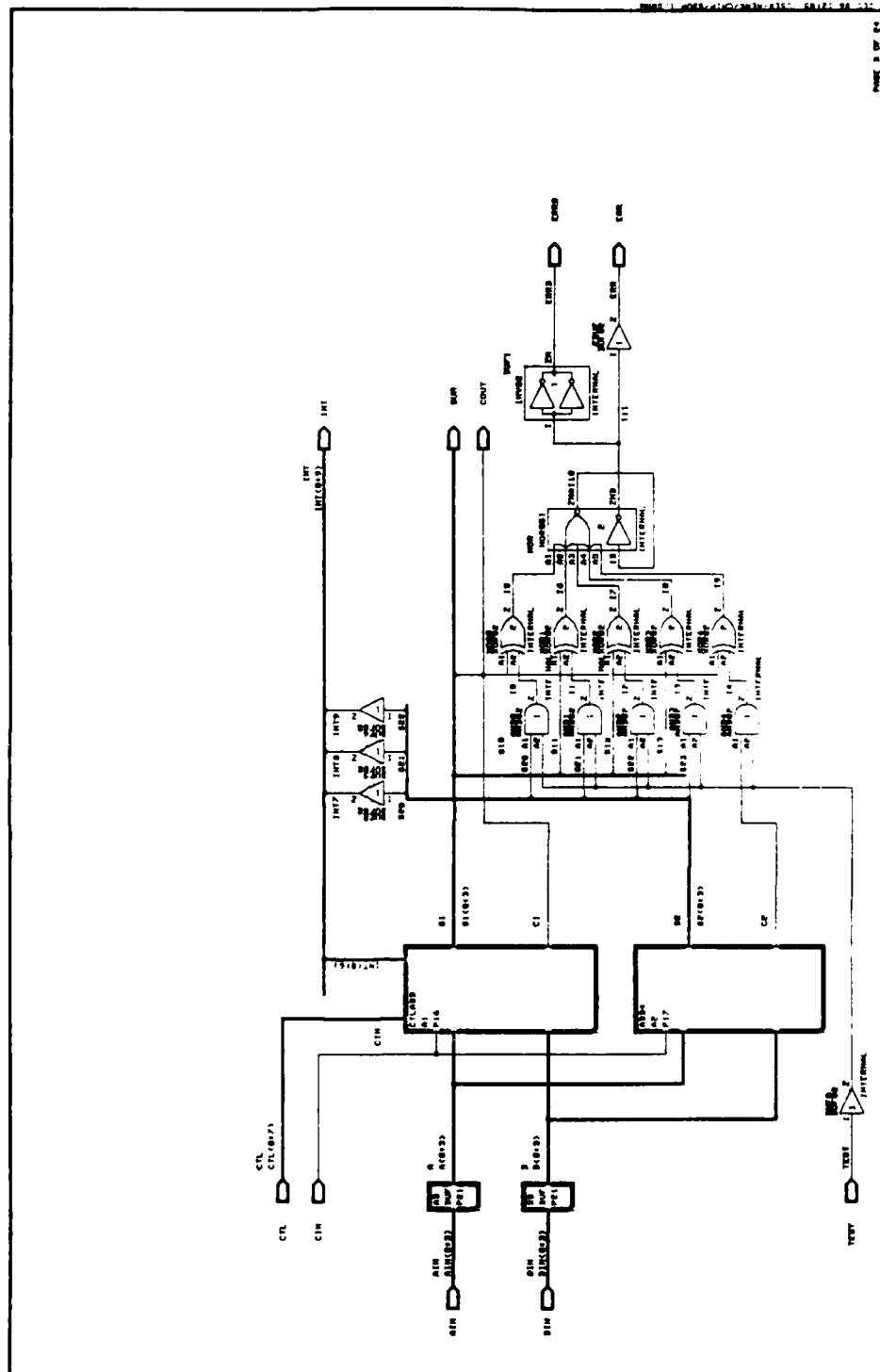


Figure 6. Simple duplication

loading on the input bus; the design of the buffers is shown in Fig. 8. Under normal operating conditions (both adders fault-free) the outputs of both adders are identical. This means that it is impossible to fully test the comparator. A stuck-at-0 output of any XOR gate will not be detected. The comparator is made testable by the addition of an AND gate to the input of each XOR gate. When the TEST line is set to 0, the XOR gates can be tested in turn by applying a 1 to one XOR gate while the other XOR inputs are set to 0. This will detect a stuck-at-0 fault on any XOR gate output.

The design of the CTLADD adder illustrated in Fig. 9 will now be discussed briefly. Each of the input lines to the full adders has an error injection circuit (shown in Fig. 1 and discussed earlier). There are also seven lines which make internal nodes observable. The choice of which nodes to observe was motivated by how much new information each node could provide. This choice was made more difficult by not knowing exactly what the results of the experiment will be.

In the case of CTLADD, four of the inputs to the full adders (just after the error injection circuitry) and three of the interstage carries are observed. All the full adder outputs are therefore directly accessible. It was argued that observation of the other full adder inputs would not provide much more information, since all the stages are identical. The remaining three lines were instead used to observe some of the SUM output lines of the duplicate adder (which would not otherwise be observable), and the three low order outputs were chosen arbitrarily.

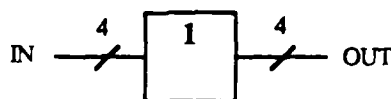


Figure 8. Input buffer

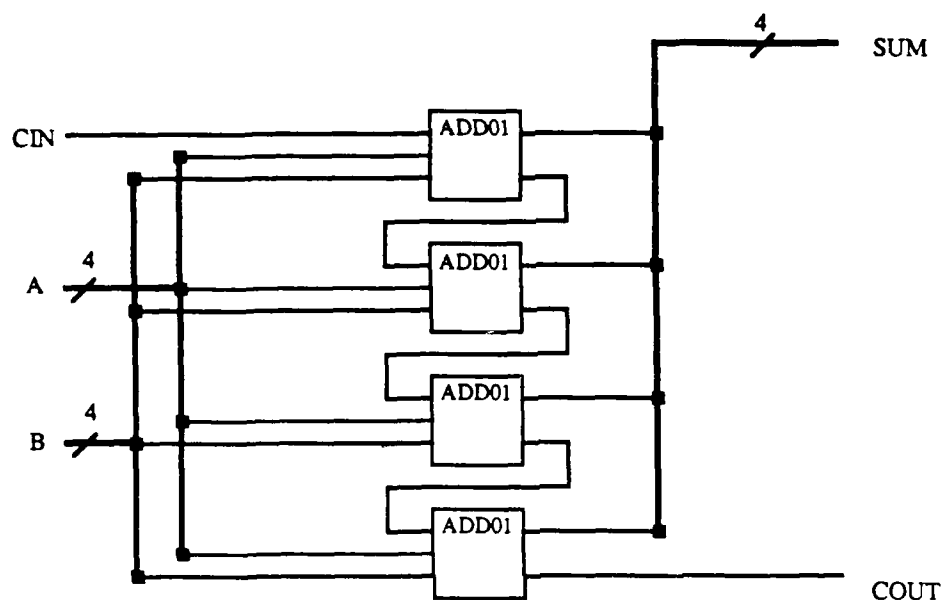


Figure 7. 4-bit adder

7.2 Dual logic implementation

A weakness of duplication for error detection is the occurrence of common mode failures. A common mode failure occurs when both circuits fail in the same way at the same time. This is very likely to happen if the fault is caused by an environmental disturbance. For VLSI the problem is especially acute since circuits are in such close physical and electrical proximity on the chip.

To combat common mode failures, some authors suggest the use of functional dual implementations [Sedmak 78]. The dual of function is obtained by exchanging all AND and OR operators [McCluskey 86]. When the inputs to the dual network are complemented, the output will be the complement of the original network output. This will reduce the probability that the circuits fail in the same way when a disturbance affects them. The design of a functional dual full adder is shown in Fig. 10.

Fig. 12 shows four of these full adders interconnected to form a TSC 4-bit adder. The complemented values of the input signals which are required by the dual full adders are generated

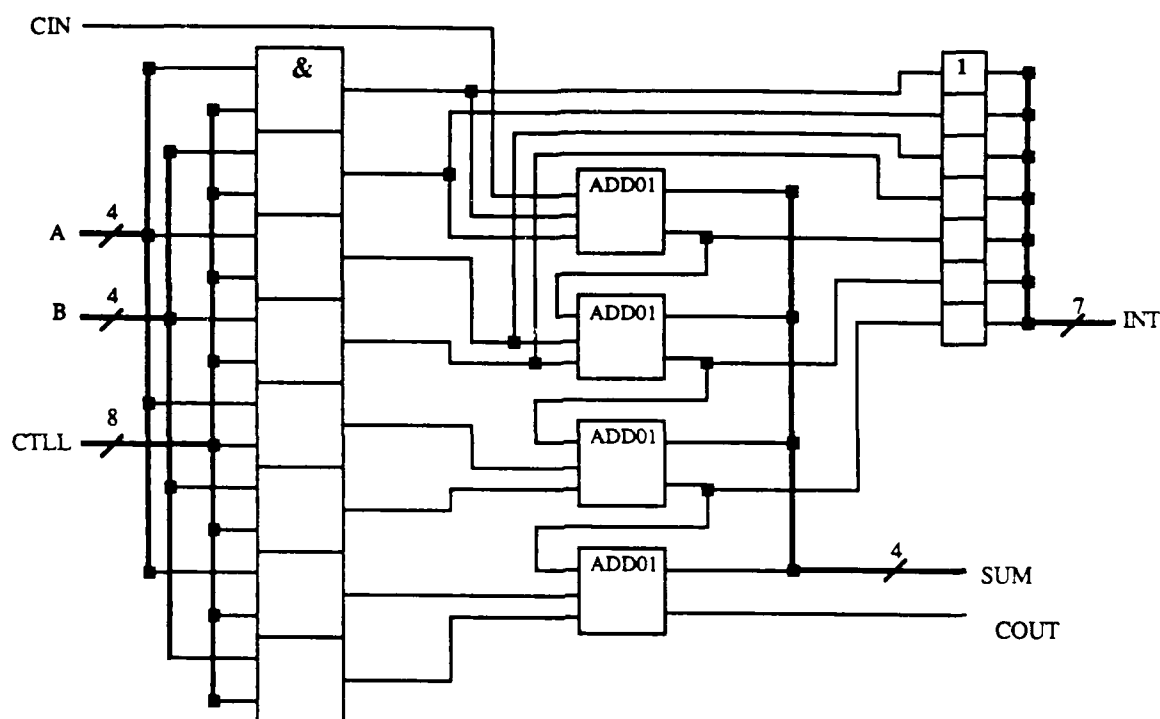


Figure 9. 4-bit adder with fault injection

locally. Each uncomplemented input of the adder has an error injection AND gate for local fault injection. Checking of the output is done by a tree of TSC two-rail checkers. The design of a TSC two-rail checker is shown in Fig. 11.

The observation of internal nodes is similar to that in SDUP. Four of the nodes are on the full adder inputs, directly after the fault injection circuitry. Three of them are the true values of the interstage carries. For the other three, the complemented value of the low-order interstage carry and the outputs of the first level low-order TSC checker were chosen. This will hopefully reveal more about the propagation of injected faults through different levels of circuitry.

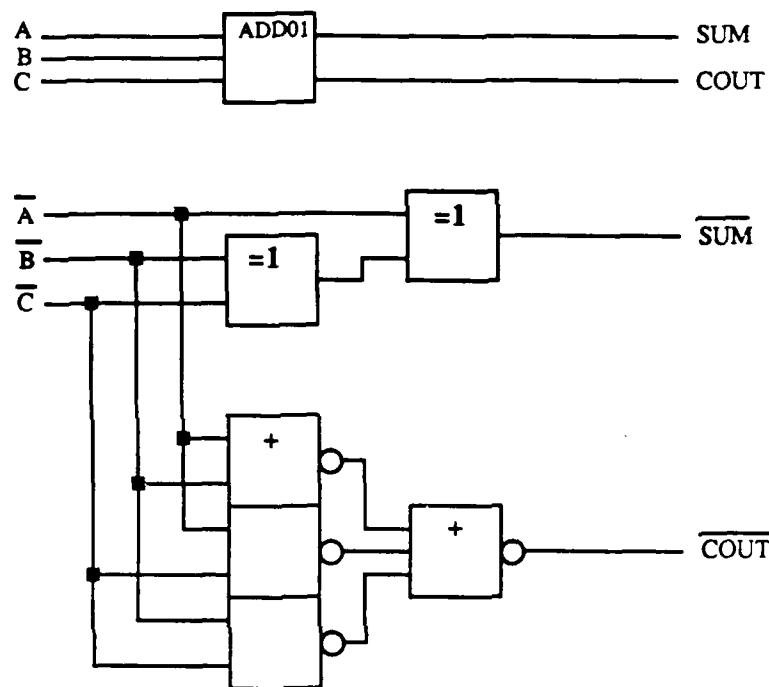


Figure 10. Dual full adder

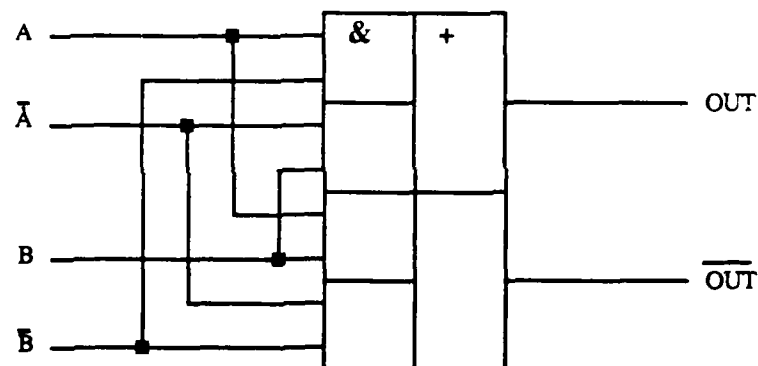


Figure 11. TSC two-rail checker

7.3 Alternative dual implementation

It has been suggested that a "different" implementation might also reduce the probability of common mode failures [Tamir 85]. In this circuit the adder is implemented differently by

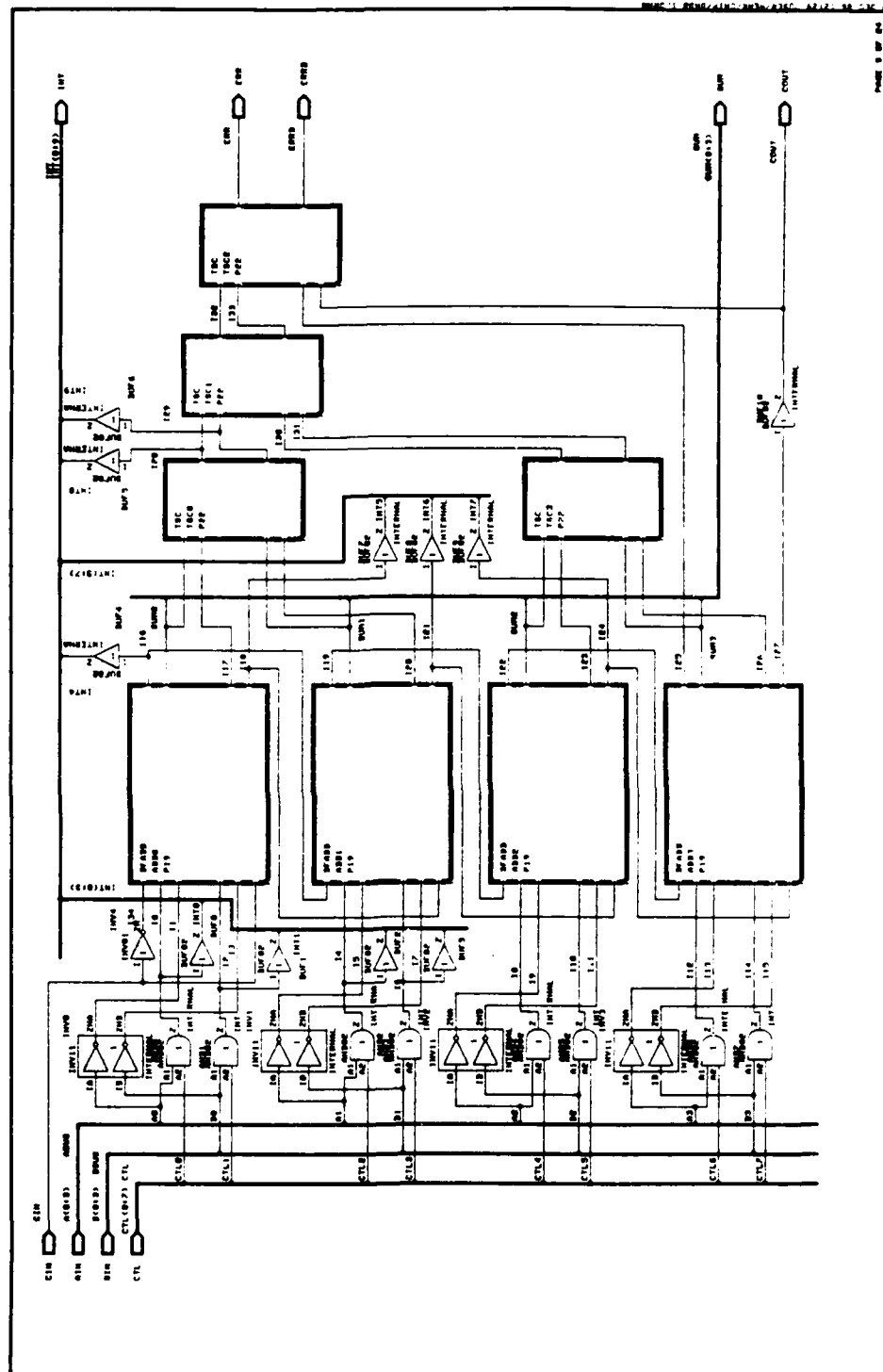


Figure 12. 4-bit dual adder

replacing the XOR gates by an AND-OR structure, and the carry circuitry by a more conventional type than that used in the dual implementation [Waser 82].

The high level structure of the alternative dual adder is once again identical to that of the functional dual adder shown in Fig. 12. The design of the alternative dual full adder is shown in Fig. 13. It should be noted that this full adder is not fault-secure for single-stuck faults, since the two adders share the uncomplemented inputs.

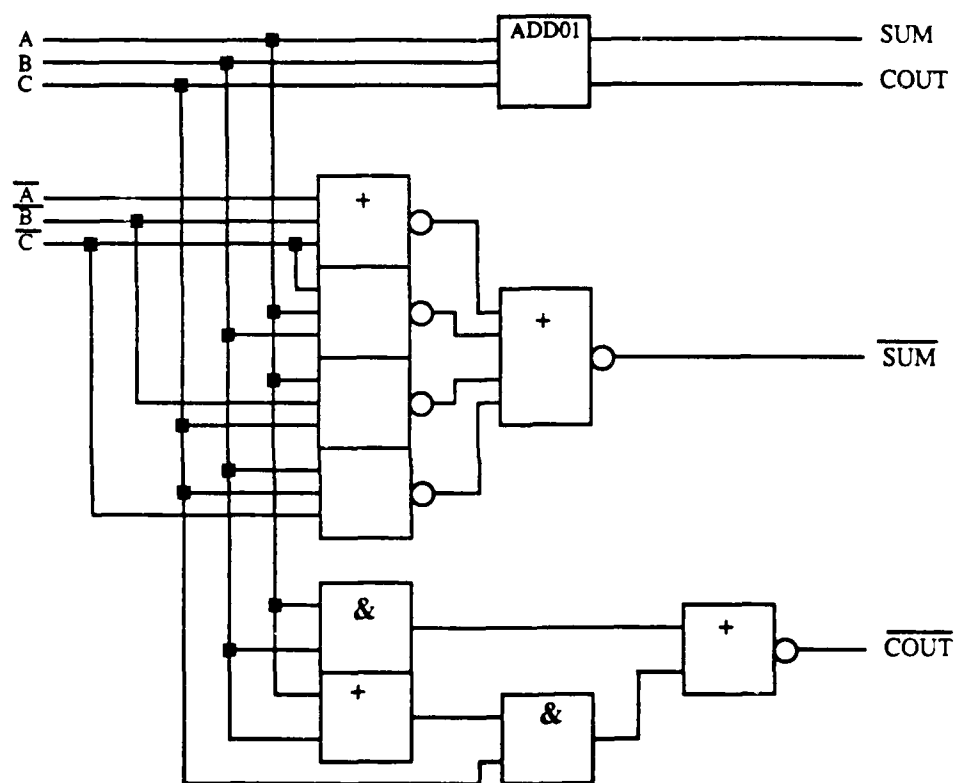


Figure 13. Alternative dual full adder

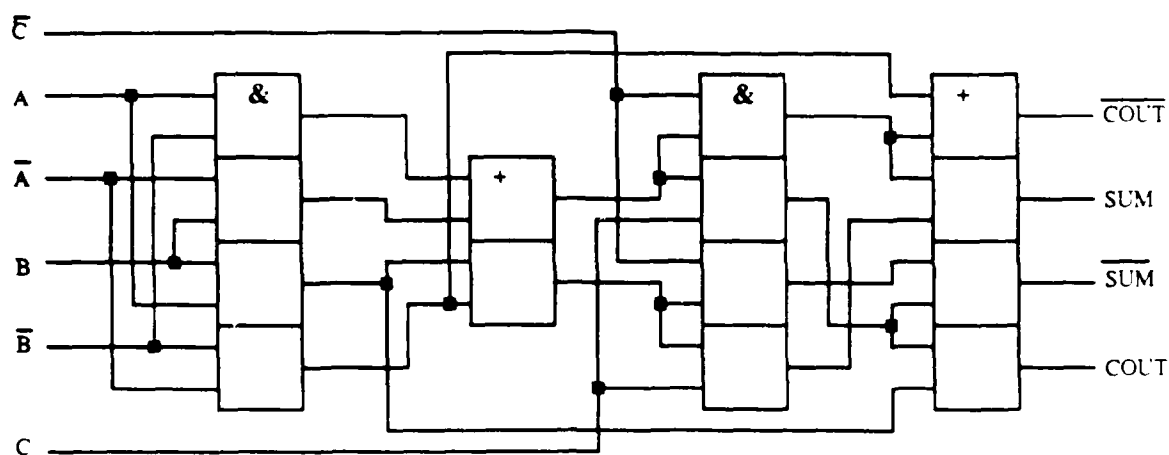


Figure 14. Two-rail full adder

7.4 TSC two-rail adder

The two-rail full adder circuit shown in Fig. 14 is suggested by Ho in his Ph.D. thesis [Ho 76]. The high level structure of the two-rail adder is identical to that of the dual adder shown in Fig. 12. The only difference between the two is in the internal design of the full adders. The observation of internal nodes is the same as in the previous scheme.

7.5 Parity prediction

Parity prediction is a well-known technique for error detection in adders [Kraft 81]. The concept has been extended to general combinational circuits by others [Khodadad-Mostashiry 79]. The basic idea is that it is possible to predict what the parity of the result of the addition should be by looking at the operands. This is done by replicating the carry circuitry, and forming the XOR of the carry bits and the parity of the two operands.

The adder with parity prediction is shown in Fig. 16. The input to each full adder has circuitry for local fault injection as before. The three level parity tree on the input lines form the combined parity of the two input numbers. There are four duplicate carry units (DUPC) which are connected to the input lines before the fault injection circuitry. This was done to

allow more experimental flexibility, since faults which also affect the duplicate carry circuitry can be injected globally. A duplicate carry unit is shown in Fig. 15.

The outputs of the duplicate carry circuits are combined by a second parity tree. The XOR of this result with the input parity is the predicted parity. Finally, the parity of the sum is formed by a third parity tree and compared to the predicted parity. The XOR gate which does the comparison is made testable by an AND gate connected to the TEST signal.

For this adder the input lines before the local fault injection circuitry are sampled. This will allow observation of the effect of global fault injection on the value of a node. It is possible that the long metal lines between the site of the fault injection and the point of observation might have an influence on the value of the node. As before, the three interstage carry signals are observed, as are the outputs of the three low-order duplicate carry units. This will once again shed light on the propagation of errors through levels of logic circuitry.

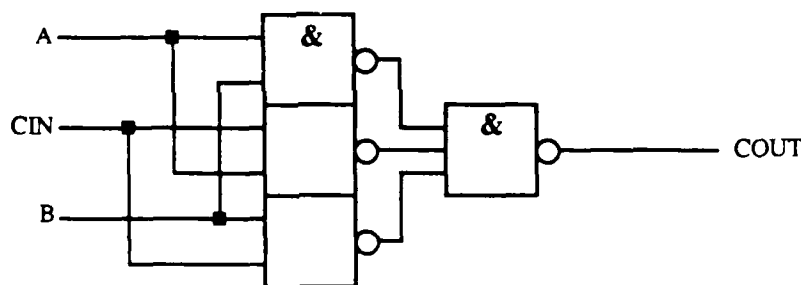


Figure 15. Duplicate carry unit

7.6 Low cost residue coding

The final scheme is a low-cost residue adder [Kraft 81]. For each operand, the residue (mod A) is calculated, where A is a number of the form $2^n - 1$, with n typically an integer much smaller than the word length of the adder. The residue (mod A) of the sum will then be equal to the residue (mod A) of the sum of the residues of the operands.

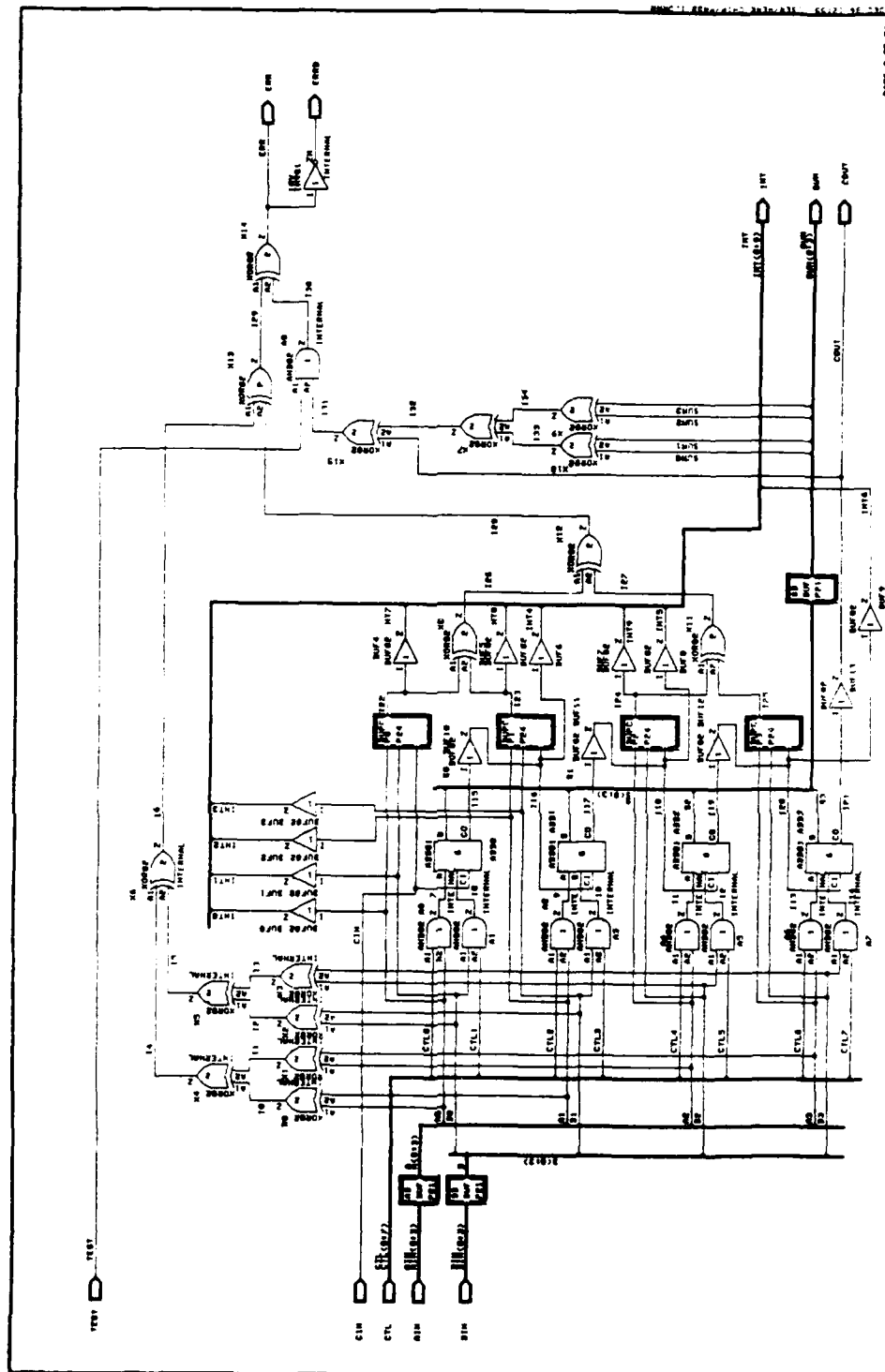


Figure 16. Adder with parity prediction

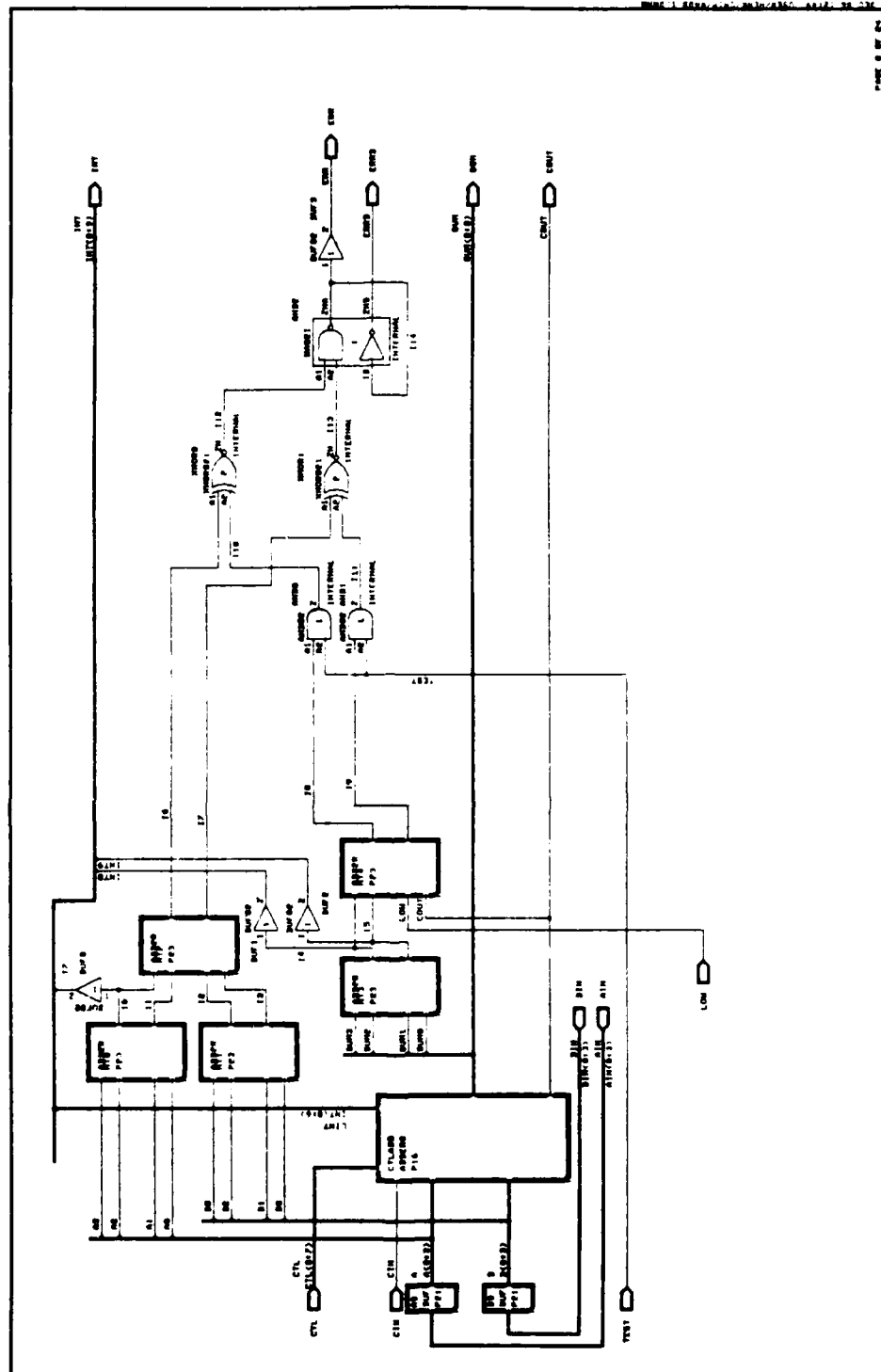


Figure 17. 4-bit adder with (mod 3) residue checking

For this experiment $n = 2$, so that checking is done by (mod 3) addition. The circuit is shown in Fig. 17. The 4-bit adder module (CTLADD) is modified for local fault injection and is identical to the one used in the simple duplication scheme and shown in Fig. 9. A tree of (mod 3) adders (module ADD2R) is used to calculate the (mod 3) residue of the two input numbers. One (mod 3) adder calculates the residue of the sum. However, there is also a carry out signal, and this has to be taken into account. A fourth (mod 3) adder adds in the carry.

The design of a (mod 3) adder is shown in Fig. 18. It is fully combinational with no end-around carry. Simulation showed that a 2-bit adder with end-around carry is prone to oscillation. This problem is also mentioned in [Wakerly 78]. An adder with end-around carry also suffers from the fact that it has two representations for zero (the all-1 and the all-0 words). This complicates the design of comparators. In this case the residues can be compared by two XOR gates. The comparators are made testable by gating one input of each XOR gate through an AND gate.

The CTLADD module has the same internal node sampling as discussed previously. An additional three nodes are sampled. Both outputs of the (mod 3) adder at the CTLADD adder output are sampled. This will shed light on the propagation of errors through multiple gates. The low-order output of the module AT0 (mod 3) adder will allow observation of the effect of a long signal run on the global fault injection.

8. DESIGN OF SUPPORT CIRCUITRY

8.1 CED schemes tri-state buffers and latches

The output latches capture all of the adder outputs and internal nodes on the falling edge of the clock (the latches are enabled when the clock signal is low, but the clock signal is inverted by the input buffers). The latch outputs are connected to the output bus via tri-state buffers (active low enable signals). The circuit is shown in Fig. 19.

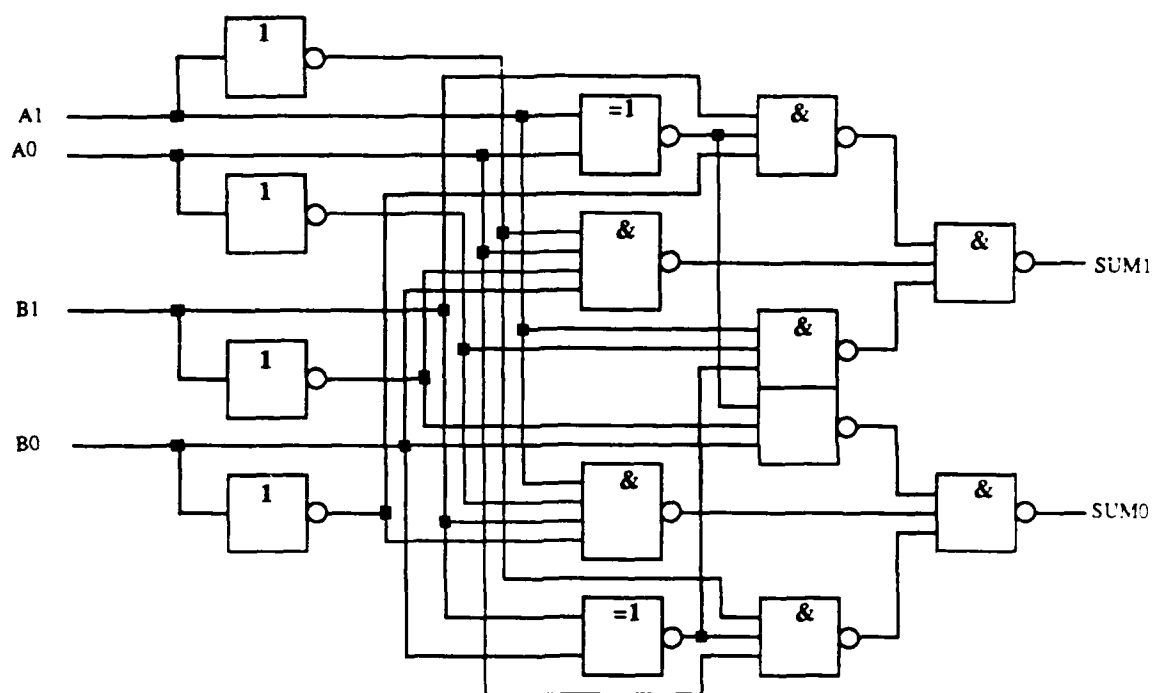


Figure 18. (mod 3) adder

8.2 Counter

Test vectors are generated by an 8-bit synchronous counter with ripple carry. The counter is shown in Fig. 20. The counter stages are negative edge-triggered JK flip-flops. Since the clock signal is inverted, the counter cycles on the rising edge of the system clock. The counter is always enabled and counting. A CLR signal is provided to reset the counter.

8.3 Buffer (CNTBUF)

CNTBUF is a set of buffers which drives the test vector output pins. It is shown in Fig. 21.

8.4 Buffer (FINBUF)

FINBUF is the set of buffers which drives the output pins of the CUT. It is shown in Fig. 22.

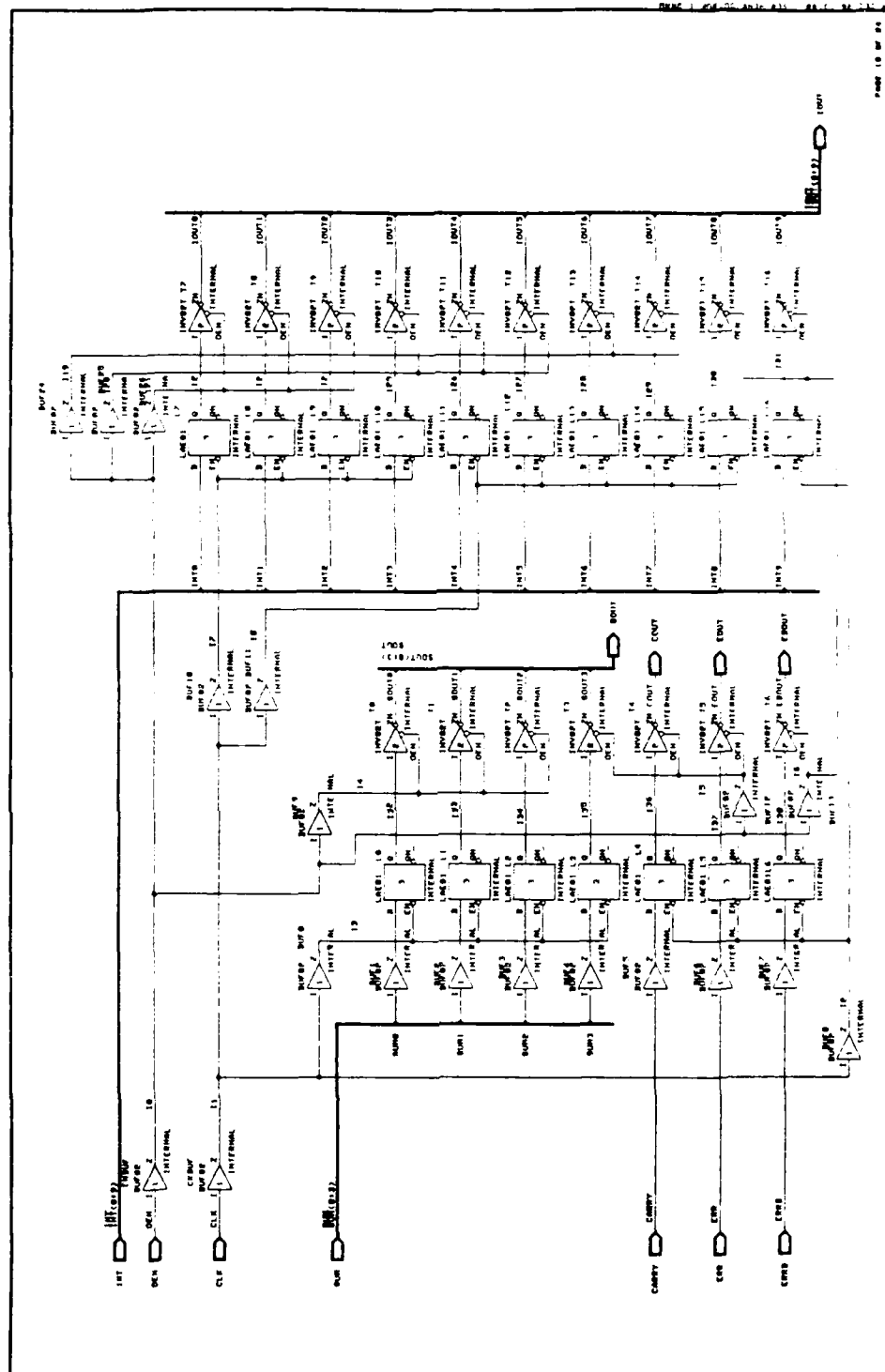


Figure 19. Output latches and tri-state buffers

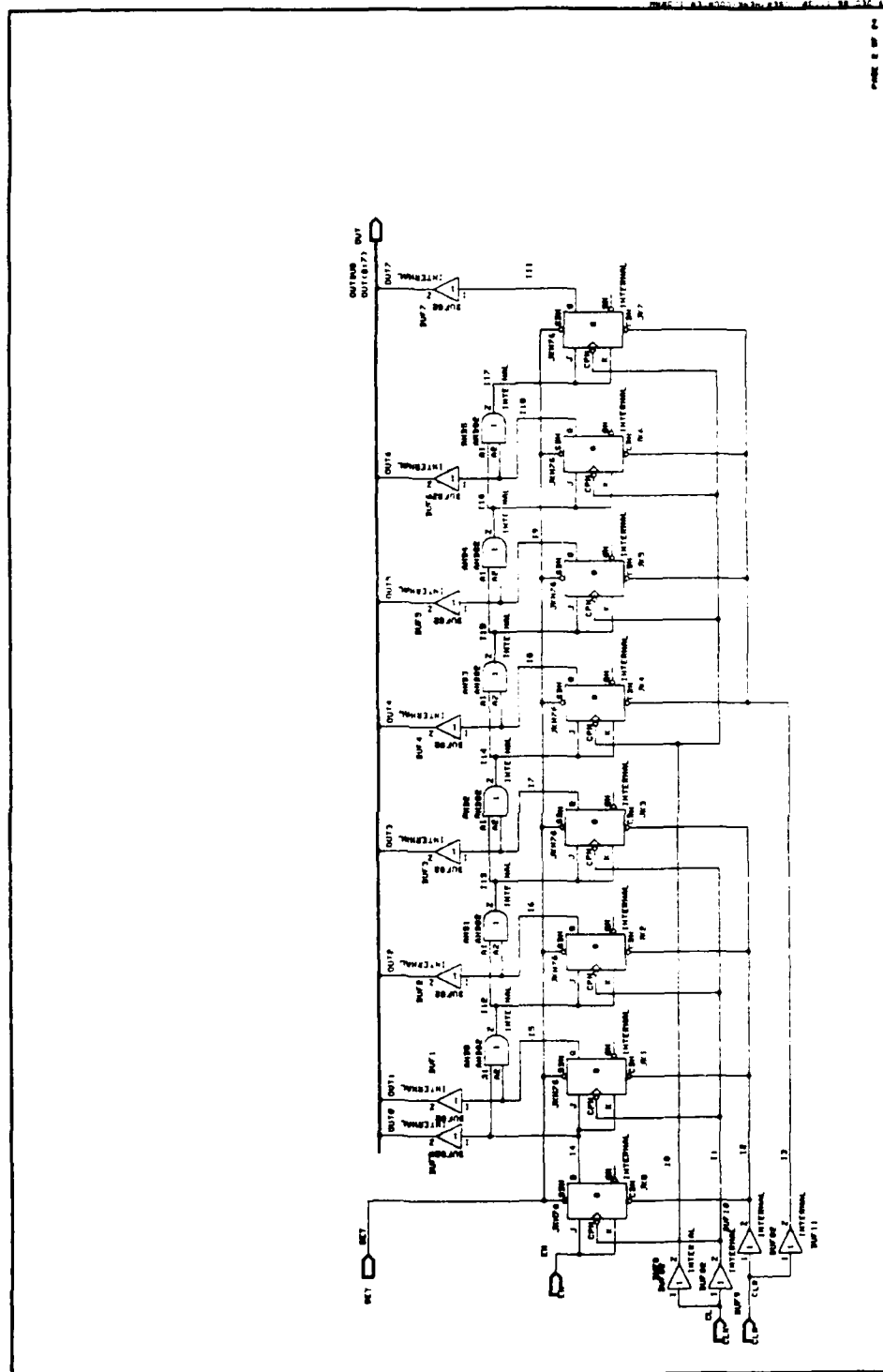


Figure 20. 8-bit counter

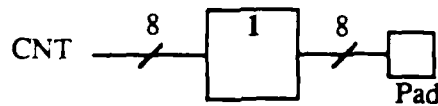


Figure 21. Counter buffer

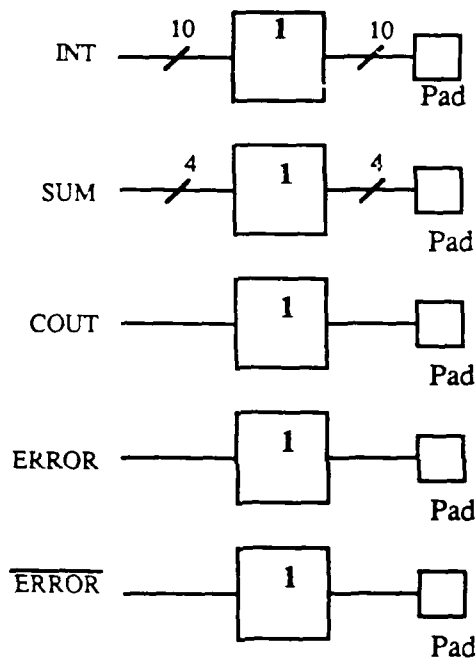


Figure 22. Final output buffer

8.5 Buffer (CTLBUF)

CTLBUF is a set of input buffers and inverters for the local fault injection control signals. It is shown in Fig. 23. The control signals are not inverted, which means the circuit will function normally when all the control signal are high. A fault is injected on a line by applying an intermediate voltage on the appropriate control line.

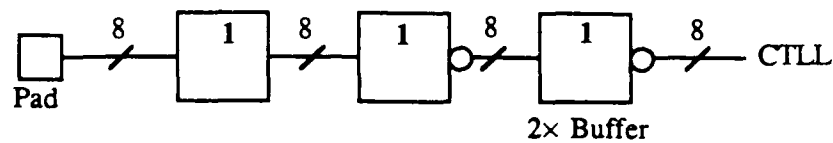


Figure 23. Local fault injection signals input buffers

8.6 Buffer (CNTIB)

CNTIB consists of input buffers for the CUT test vectors. It also has an AND gate on every line for the injection of weak input faults on the data bus. The circuit will function normally when all the control signals are high. The circuit is shown in Fig. 24.

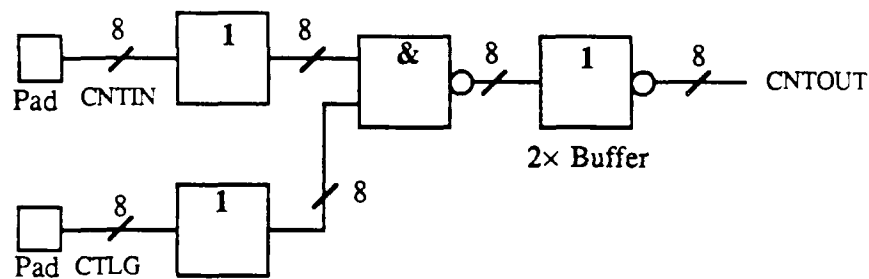


Figure 24. Test vector input buffer and global fault injection

8.7 Reference adder

The reference adder employs CED to increase confidence in the results. It has duplicated 4-bit adders (ADD4 in Fig. 7) with matching. The circuit is shown in Fig. 25.

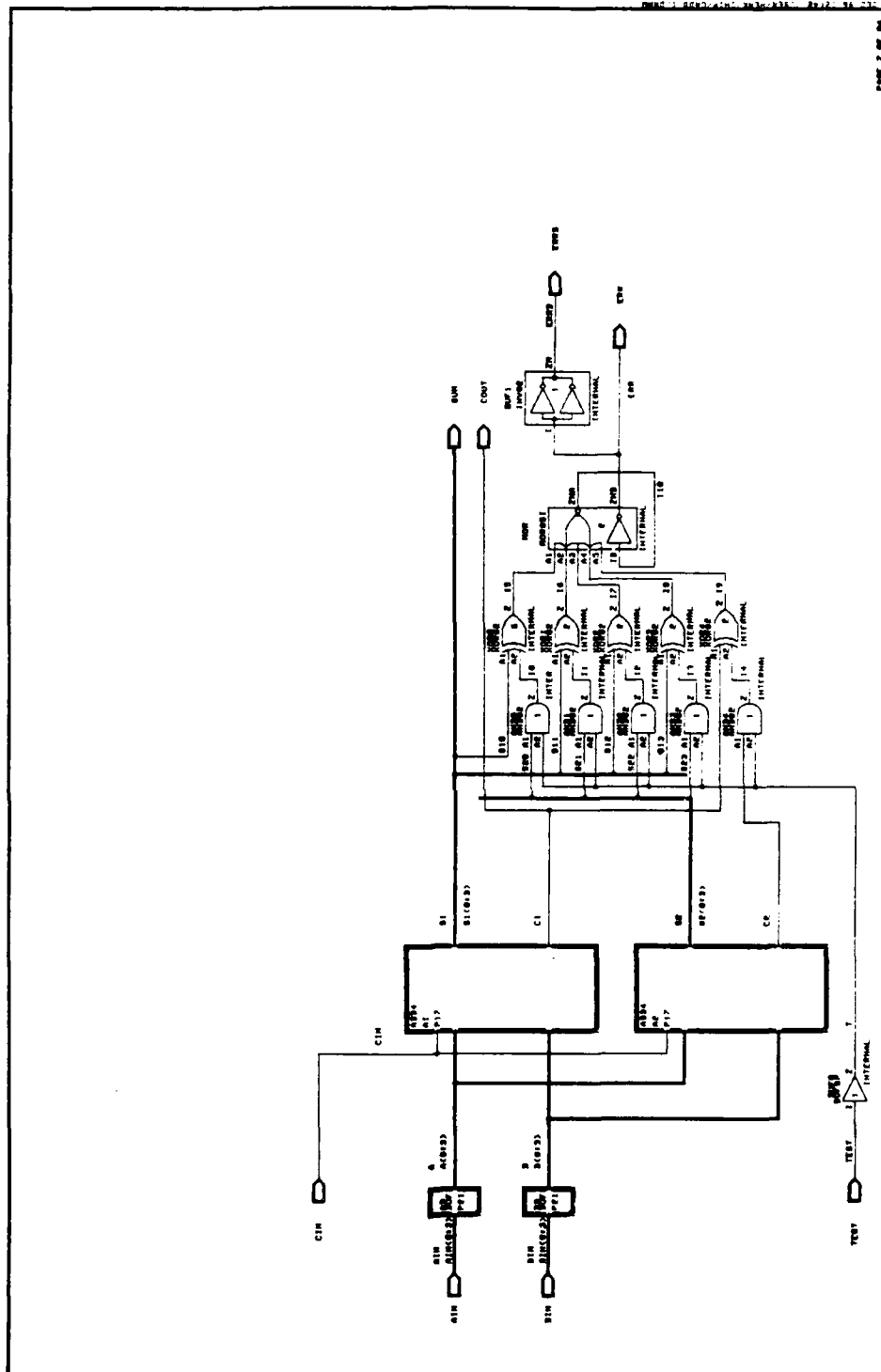


Figure 25. Reference adder

8.8 Comparator (SYSCMP)

The comparator monitors the CUT output and provides signals indicating the status of the reference adder and CED scheme under test. The outputs of the reference adder are latched to correspond to the CED scheme outputs. The circuit compares the reference sum and CUT sum and indicates the result on the ERROR signal line. The correct operation of a regular CED scheme is indicated by the REGOK signal.

$$REGOK = (ERROR \oplus ERRIN)'$$

The correct operation of a two-rail CED scheme is indicated by the TROK signal.

$$TROK = ((ERRIN \oplus ERRBIN)' \oplus ERROR)'$$

The design of the comparator is shown in Fig. 26.

ACKNOWLEDGEMENTS

This work was supported in part by the Council for Scientific and Industrial Research, South Africa, and in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization administered through the Office of Naval Research under contract No. N00014-85-K-0600. Schematic capture and logic simulation were done on a Megalogician workstation donated by Daisy Systems Corporation, Mountain View, California. The circuit was fabricated and donated by Fairchild Gate Array of Milpitas, California. The authors wish to thank Dr. Rodolfo Betancourt, Mr. Thomas Ngo, and Mr. Mick O' Brien of Fairchild for their advice and assistance with this project. The authors would also like to thank Dr. Hassanein H. Amer, Sharon Garner, Sandy Goosen, Dr. Dick L. Liu, Dave McCluskey, Prof. Takashi Nanya, and Prof. Garth Saloner for their valuable comments and suggestions.

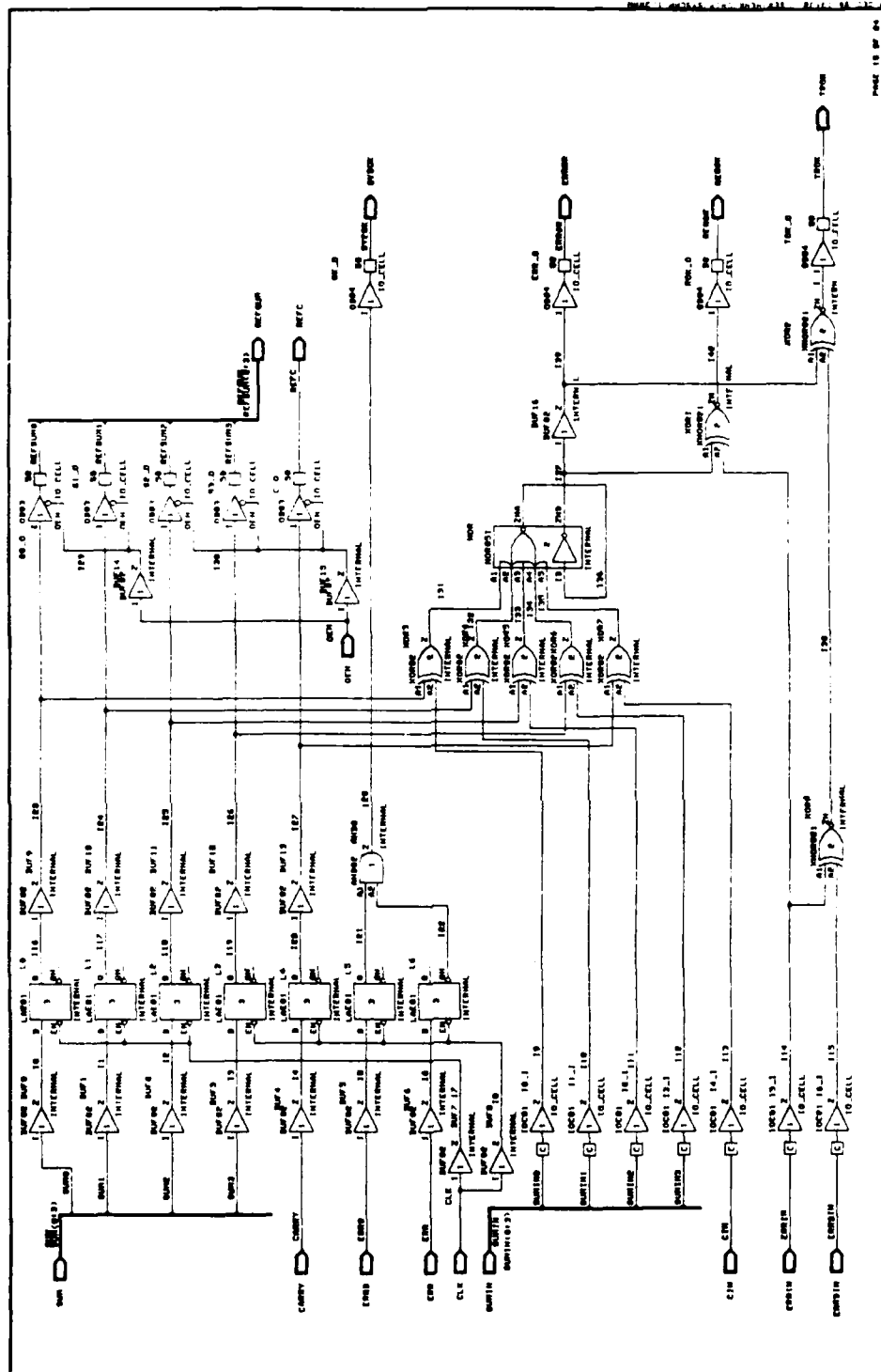


Figure 26. System comparator

REFERENCES

- [Amer 87] Amer, H. H., M. L. Cortes and E. J. McCluskey, "The Inadequacy of Conventional Dynamic Recovery Mechanisms is the Presence of Temporary Failures," CRC TR 87-11, Stanford University, Mar. 1987.
- [Carter 64] Carter, W. C. et al, "Design of serviceability features for the IBM System/360," *IBM Journal of Research and Development*, Vol. 8, No. 2, pp. 115-126, Apr. 1964.
- [Cortés 87] Cortés, M. L., *Temporary Failures in Digital Circuits: Experimental Results and Fault Modelling*, Ph.D. Thesis, Stanford University, Mar. 1987.
- [Cortés 86a] Cortés, M. L. et al, "Modeling Power Supply Disturbances in Digital Circuits," *IEEE International Solid-State Circuits Conference*, pp. 164-165, Feb. 1986.
- [Cortés 86b] Cortés, M. L. et al, "Properties of Transient Errors Due to Power Supply Disturbances," *IEEE International Symposium on Circuits and Systems*, pp. 1046-1049, May 1986.
- [Cortés 86c] Cortés, M. L. and E. J. McCluskey, "An Experiment on Intermittent-Failure Mechanisms," *IEEE International Test Conference*, pp. 435-442, Sept. 1986.
- [Crouzet 82] Crouzet, Y, and B. Decouty, "Measurement of Fault Detection Mechanisms," *Proc. FTCS 12*, pp. 373-376, 1982.
- [Ho 76] Ho, D. S-M., *The Design of Totally Self-Checking Systems*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1976.
- [Khodadad-Mostashiry 79] Khodadad-Mostashiry, B., "Parity Prediction in Combinational Circuits," Technical Note No. 151, CRC, Stanford University, 1979.
- [Kraft 81] Kraft, G. D. and W. N. Toy, *Microprogrammed Control and the Reliable Design of Small Computers*, Prentice-Hall, Englewood Cliffs NJ, 1981.
- [McCluskey 86] McCluskey, E. J., *Logic Design Principles*, Prentice-Hall, Englewood Cliffs NJ, 1986.
- [McConnel 79] McConnel, S. R., D. P. Siewiorek and M. M. Tsao, "The Measurement and Analysis of Transient Errors in Digital Computer Systems," *Proc. FTCS 9*, pp. 67-70, June 1979.
- [Schuette 86] Schuette, M. A. et al, "Experimental Evaluation of Two Concurrent Error Detection Schemes," *Proc. FTCS 16*, pp. 138-143, 1986.
- [Sedmak 78] Sedmak, R. M. and H. L. Liebergot, "Fault Tolerance of a General Purpose Computer Implemented by VLSI," *Proc. FTCS 8*, pp. 137-143, 1978.
- [Tamir 85] Yuval Tamir, *Fault Tolerance for VLSI Multicomputers*, Ph.D. Thesis, University of California, Berkeley, 1985.
- [Wakerley 78] Wakerley, J. F., *Error-Detecting Codes, Self-Checking Circuits and Applications*, North-Holland, New York, 1978.
- [Waser 81] Waser, S., and M. J. Flynn, *Introduction to Arithmetic for Digital Systems Designers*, Holt, Rinehart and Winston, 1981.

APPENDIX A: DRAWING BLOCK AND SIGNAL NAMES

Block names

CADD	reference adder
CNTBUF	buffer to drive output pins
CNTIB	input buffer with error injection circuitry
CNT8	8-bit counter
CTLBUF	input buffer for local error injection
DADD	adder with dual
D2BUF	tri-state buffer
FINBUF	buffer driving output pins
PADD	adder with parity prediction
PBUF	tri-state buffer
RADD	low-cost residue adder
RBUF	tri-state buffer
SDUP	simple duplication
SYSCMP	comparator
S2BUF	tri-state buffer
TRADD	two-rail adder
T2BUF	tri-state buffer
WADD	adder with alternative dual implementation
W2BUF	tri-state buffer

Input signal names

CEN	enable for reference adder output
CIN	carry input for comparator
CLK	clock signal
CLR	reset the counter
CNTIN	input for CED adders

CTLG	global error injection control
CTLL	local error injection control
DEN	dual tri-state enable
ERRBIN	complementary error input for comparator
ERRIN	error input for comparator
PEN	parity prediction tri-state enable
REN	residue code tri-state enable
SEN	single duplication tri-state enable
SUMIN	sum input for comparator
TEN	two-rail tri-state enable
TEST	test mode
WEN	alternative dual tri-state enable

Output signal names

CNTOUT	output of counter generating test vectors
ERROR	disagreement between reference sum and sum from CUT
OUTC	carry output of CUT
OUTERR	error detected in CUT
OUTERRB	complement of OUTERR
OUTINT	internal nodes in CUT (10 bits)
OUTSUM	sum output of CUT (4 bits)
REFC	reference carry output
REFSUM	reference sum output (4 bits)
REGOK	regular scheme functioning correctly
SYSOK	reference adder function correctly
TROK	two-rail scheme functioning correctly

APPENDIX B: OBSERVABLE INTERNAL NODES

Adder with simple duplication

The following internal nodes are observable:

- INT0 A0 input on CTLADD (after error injector)
- INT1 B0 input on CTLADD (after error injector)
- INT2 A1 input on CTLADD (after error injector)
- INT3 B1 input on CTLADD (after error injector)
- INT4 interstage carry on CTLADD (from stage 0 to stage 1)
- INT5 interstage carry on CTLADD (from stage 1 to stage 2)
- INT6 interstage carry on CTLADD (from stage 2 to stage 3)
- INT7 stage 0 sum on ADD4
- INT8 stage 1 sum on ADD4
- INT9 stage 2 sum on ADD4

Two-rail, dual, and alternative dual adders

- INT0 A0 input (after error injector)
- INT1 B0 input (after error injector)
- INT2 A1 input (after error injector)
- INT3 B1 input (after error injector)
- INT4 interstage (carry)' (from stage 0 to stage 1)
- INT5 interstage carry (from stage 0 to stage 1)
- INT6 interstage carry (from stage 1 to stage 2)
- INT7 interstage carry (from stage 2 to stage 3)
- INT8 top output (I28) of first TSC checker (TSC0) in tree
- INT9 bottom output (I29) of first TSC checker (TSC0) in tree

Adder with parity prediction

INT0 A0 input on duplicate carry (before error injector)
INT1 B0 input on duplicate carry (before error injector)
INT2 A1 input on duplicate carry (before error injector)
INT3 B1 input on duplicate carry (before error injector)
INT4 interstage carry (from stage 0 to stage 1)
INT5 interstage carry (from stage 1 to stage 2)
INT6 interstage carry (from stage 2 to stage 3)
INT7 stage 0 duplicate carry
INT8 stage 1 duplicate carry
INT9 stage 2 duplicate carry

Adder with residue code

INT0 A0 input on CTLADD (after error injector)
INT1 B0 input on CTLADD (after error injector)
INT2 A1 input on CTLADD (after error injector)
INT3 B1 input on CTLADD (after error injector)
INT4 interstage carry on CTLADD (from stage 0 to stage 1)
INT5 interstage carry on CTLADD (from stage 1 to stage 2)
INT6 interstage carry on CTLADD (from stage 2 to stage 3)
INT7 bit 0 of (mod 3) adder for A operand
INT8 bit 0 of (mod 3) adder for sum
INT9 bit 1 of (mod 3) adder for sum

APPENDIX C: FAULT INJECTION CONTROL

The principle behind fault injection is explained in Section 3. The system will operate normally when all the fault injection control signals are high. A fault is injected on a specific line by applying an intermediate voltage ("weak input") to the appropriate fault injection control line. The following tables associate control lines with data lines.

Global Fault Injection

Control Signal	Adder Input	Pin Number
CTLG0	A0	M11
CTLG1	A1	L10
CTLG2	A2	N12
CTLG3	A3	N11
CTLG4	B0	M10
CTLG5	B1	L9
CTLG6	B2	N10
CTLG7	B3	M9

Local Fault Injection

Control Signal	Adder Input	Pin Number
CTLL0	A0	M6
CTLL1	A1	L6
CTLL2	A2	N5
CTLL3	A3	M5
CTLL4	B0	N4
CTLL5	B1	L5
CTLL6	B2	M4
CTLL7	B3	N3

APPENDIX D: PACKAGE DETAIL

The chip is packaged in a 121-pin ceramic pin-grid array.

Complete pinout

Pin Number	Die Pad	I/O Type	Signal Name
A1	1	V _{ss}	
A2	4	n.c.	
A3	5	n.c.	
A4	8	n.c.	
A5	10	n.c.	
A6	13	n.c.	
A7	14	n.c.	
A8	17	n.c.	
A9	20	n.c.	
A10	22	n.c.	
A11	25	n.c.	
A12	28	n.c.	
A13	31	V _{dd}	
B1	118	n.c.	
B2	119	out	CNTOUT0
B3	2	n.c.	
B4	6	in	CLR
B5	9	n.c.	
B6	12	in	CLK
B7	16	n.c.	
B8	18	n.c.	
B9	21	n.c.	
B10	24	n.c.	
B11	26	n.c.	
B12	29	n.c.	
B13	34	in	SEN
C1	115	n.c.	
C2	116	out	CNTOUT2
C3	120	V _{dd}	
C4	3	n.c.	
C5	7	n.c.	
C6	11	n.c.	
C7	15	n.c.	
C8	19	n.c.	
C9	23	n.c.	
C10	27	n.c.	
C11	30	V _{ss}	
C12	32	n.c.	
C13	35	out	INT1
D1	112	n.c.	
D2	114	out	CNTOUT3
D3	117	out	CNTOUT1
D4	none	n.c.	
D11	33	out	INT0
D12	36	OUT	INT2

D13	38	out	INT3
E1	110	out	CNTOUT6
E2	111	out	CNTOUT5
E3	113	out	CNTOUT4
E11	37	in	TEN
E12	39	out	INT4
E13	40	n.c.	
F1	107	out	REFSUM0
F2	108	out	CNTOUT7
F3	109	in	TEST
F11	41	out	INT5
F12	42	out	INT6
F13	43	in	PEN
G1	104	out	REFSUM2
G2	106	in	CEN
G3	105	out	REFSUM1
G11	45	out	INT8
G12	46	in	REN
G13	44	out	INT7
H1	103	n.c.	
H2	102	out	REFSUM3
H3	101	out	REFC
H11	49	in	DEN
H12	48	out	SUMOUT0
H13	47	out	INT9
J1	100	in	SUMIN0
J2	99	out	SYSOK
J3	97	in	SUMIN1
J11	53	out	SUMOUT3
J12	51	out	SUMOUT2
J13	50	out	SUMOUT1
K1	98	out	ERROR
K2	96	out	REGOK
K3	93	in	SUMIN3
K11	57	out	ERRBOUT
K12	54	out	OUTC
K13	52	n.c.	
L1	95	out	TROK
L2	92	n.c.	
L3	90	V _{ss}	
L4	87	in	ERRIN
L5	83	in	CTLL5
L6	79	in	CTLL1
L7	75	in	CNTIN5
L8	71	in	CNTIN1
L9	67	in	CTLG5
L10	63	in	CTLG1
L11	60	V _{dd}	
L12	56	out	OUTERR
L13	55	in	WEN
M1	94	in	SUMIN2
M2	89	n.c.	
M3	86	in	ERRBIN

M4	84	in	CTLL6
M5	81	in	CTLL3
M6	78	in	CTLL0
M7	76	in	CNTIN6
M8	72	in	CNTIN2
M9	69	in	CTLG7
M10	66	in	CTLG4
M11	62	in	CTLG0
M12	59	n.c.	
M13	58	n.c.	
N1	91	V _{dd}	
N2	88	in	CIN
N3	85	in	CTLL7
N4	82	in	CTLL4
N5	80	in	CTLL2
N6	77	in	CNTIN7
N7	74	in	CNTIN4
N8	73	in	CNTIN3
N9	70	in	CNTIN0
N10	68	in	CTLG6
N11	65	in	CTLG3
N12	64	in	CTLG2
N13	61	V _{ss}	

END

DATE

FILMED

5-88

DTIC